QuantifiedSelf App V2

Problem definition

Modern Application Development - 2

Last Updated: 2022-04-31

Frameworks to be used

- Flask for API
- VueJS for UI
- VueJS Advanced with CLI (only if required, not necessary)
- Jinja2 templates if required
    - Not to be used for UI
- Bootstrap etc., if required
- SQLite for database
- Redis for caching
- Redis and Celery for batch jobs
- It should be possible to run all the demos on the student's computer, which should either be a Linux based system or should be able to simulate the same. You can use WSL for Windows OS.

QuantifiedSelf

- Used for self tracking - tracking habits, activities, other life parameters etc
- User can have multiple trackers
- Each tracker will have a
  - id
  - Name
  - Description
  - Tracker type
  - Settings
- User can log to one more tracker at any time, each time it’s logged it will capture
  - TimeStamp
  - Tracker
  - Value (based on the corresponding tracker type)
  - Note
- System will track progress over time and shows graphs trend lines etc

Terminology

- Tracker - Corresponding to the
- TrackerType - Type says what data is captured
  - Numerical
  - Multiple Choice
  - Time Duration
  - Boolean
- Logging - Logging an event to a tracker by providing values
- Trendline - Shows the list of logged events and may be graphs

Example - Temperature Tracker

- Can be used to log daily temperature by covid patients
- Tracker
    - Id : PK : tracker1
    - Name: Temperature
    - Description: Tracking body temperature in Fahrenheit
    - TrackerType: Numerical

- Folks can log at any time of the day.
- Example log 1
    - TimeStamp : "2022-05-26T11:42:00.73+05:30"
    - Tracker: tracker1
    - Value : 98.3
    - Note : I was feeling okay
- Example 2
    - TimeStamp : "2022-05-27T10:42:00.73+05:30"
    - Tracker: tracker1
    - Value : 100.1
    - Note : Feeling tired and bit feverish

Example - Running Tracker

- Can be used to log daily running by anyone
- Tracker
    - Id : PK : tracker2
    - Name: Running
    - Description: Tracking daily running in kilometers
    - TrackerType: Numerical

- Folks can log at any time of the day.
- Example log 1
    - TimeStamp : "2022-05-26T11:42:00.73+05:30"
    - Tracker: tracker2
    - Value : 5
    - Note : It was a good run. Felt a little tired but okay.
- Example 2
    - TimeStamp : "2022-05-27T10:42:00.73+05:30"
    - Tracker: tracker2
    - Value : 2
    - Note : Couldn't run much today
- Example 2
    - TimeStamp : "2022-05-27T18:42:00.73+05:30"
    - Tracker: tracker2
    - Value : 3
    - Note : Making up because couldn’t run in the morning

Core Functionality

- This will be graded
- Base requirements:
    - UI with Vue and Vue Components
        - User login
        - Dashboard and Trendlines
        - Tracker management
        - Logging
    - Backend Jobs
        - Export Jobs
        - Reporting Jobs
        - Alert Jobs
    - UI and Backend Performance

Core - User Login

- Form for username and password
- Use Flask Security and Token Based Authentication
- Suitable model for user

Core - Dashboard

- Dashboard with list of trackers
- Time of last review, value on tracker
- Ability to go to logging view for any tracker
- Ability to go to create or edit tracker
- Ability to go the specific tracker details

Core - Tracker management

- Create a new tracker
  - Add trackers to dashboard
- Edit a tracker
- View Tracker
  - View tracker
  - View logs related to that tracker
  - View stats and trendlines
- Remove a tracker
- Export option is required

Core - Logging

- Click on a tracker, then log the values
  - Based on the TrackerType it should show the options to log
    - Numerical - Show the text box that takes numerical values only
    - MultipleChoice - present the options
    - Time Duration - Give the ability to select time duration, like 30 mins, 1 hr 29 mins etc.
  - The current timestamp needs to be picked up automatically. But the user should have the ability to edit
- Edit a log
  - Change value, timestamp or associate notes
- Remove a log
- Export option is required

Core - Daily Reminder Jobs

- Scheduled Job - Daily reminders on Google Chat using webhook or SMS or Email
    - In the evening, every day (you can choose time of your choice)
    - Check if the user has logged events to any tracker
    - If not logged, then send the alert asking them to log

Core - Scheduled Job - Monthly Progress Report

- Scheduled Job - Monthly Progress Report
  - Come Up with a monthly progress report in HTML (email)
    - Could be for one of the tracker or summary of all
  - On the first day of the month
    - Start a job
    - Create a report
    - Send it as email

Core - User Triggered Async Job - Export as CSV

- User Triggered Async Job - Export as CSV
  - Come up with an export CSV format for Tracker and Events
  - Have a dashboard where the user can export
  - Trigger a batch job, send an alert once done

Core - Performance and Caching

- Add caching where required to increase the performance
- Add cache expiry
- UI and API Performance

Recommended (graded)

- Backend Jobs
  - Import Jobs
- Well designed PDF reports (User can choose between HTML and PDF reports)
- Single Responsive UI for both Mobile and Desktop
  - Unified UI that works across devices
  - Add to desktop feature

Optional

- Styling and Aesthetics

Evaluation

- Report (not more than 2 pages) describing models and overall system design
  - Include as PDF inside submission folder
- All code to be submitted on portal
- A brief (2-3 minute) video explaining how you approached the problem, what you have implemented, and any extra features
  - This will be viewed during or before the viva, so should be a clear explanation of your work
- Viva: after the video explanation, you are required to give a demo of your work, and answer any questions
  - This includes making changes as requested and running the code for a live demo
  - Other questions that may be unrelated to the project itself but are relevant for the course

Instructions

- This is a live document and will be updated with more details and FAQs, as we proceed.
- We will freeze the problem statement on or before 7th June, beyond which any modifications to the statement will be communicated via proper announcements.
- The project has to be submitted as a single zip file.